

Fig. 1

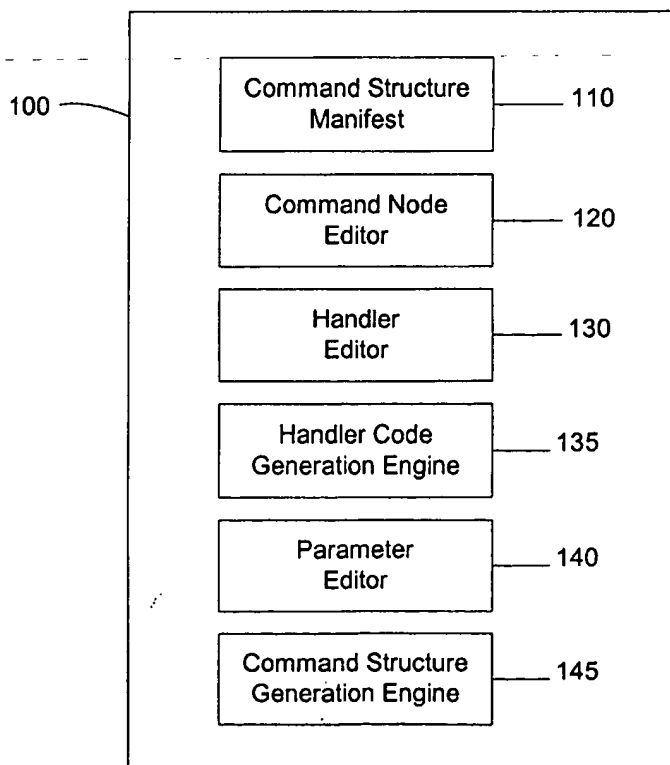


Fig. 2

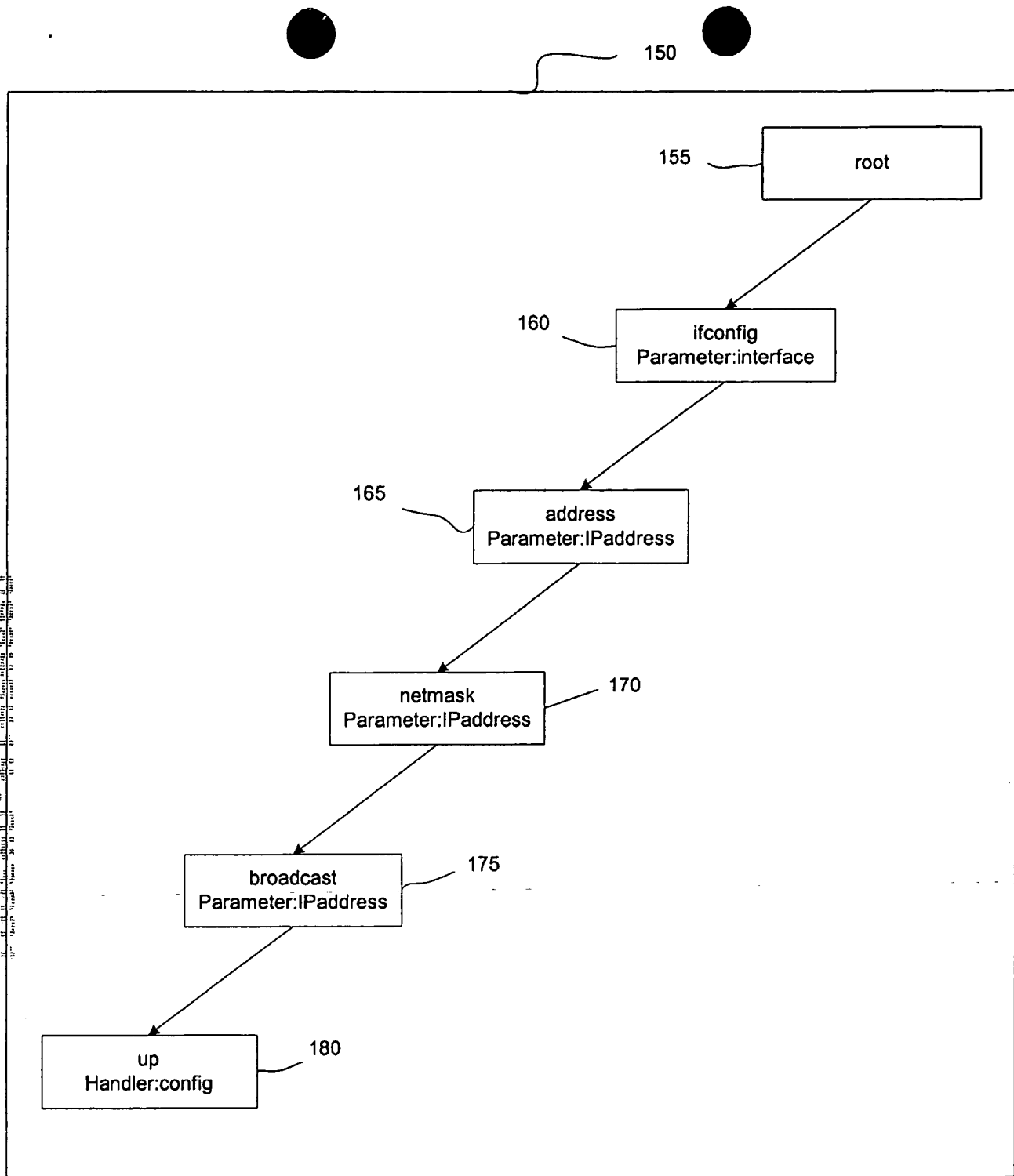


Fig. 3

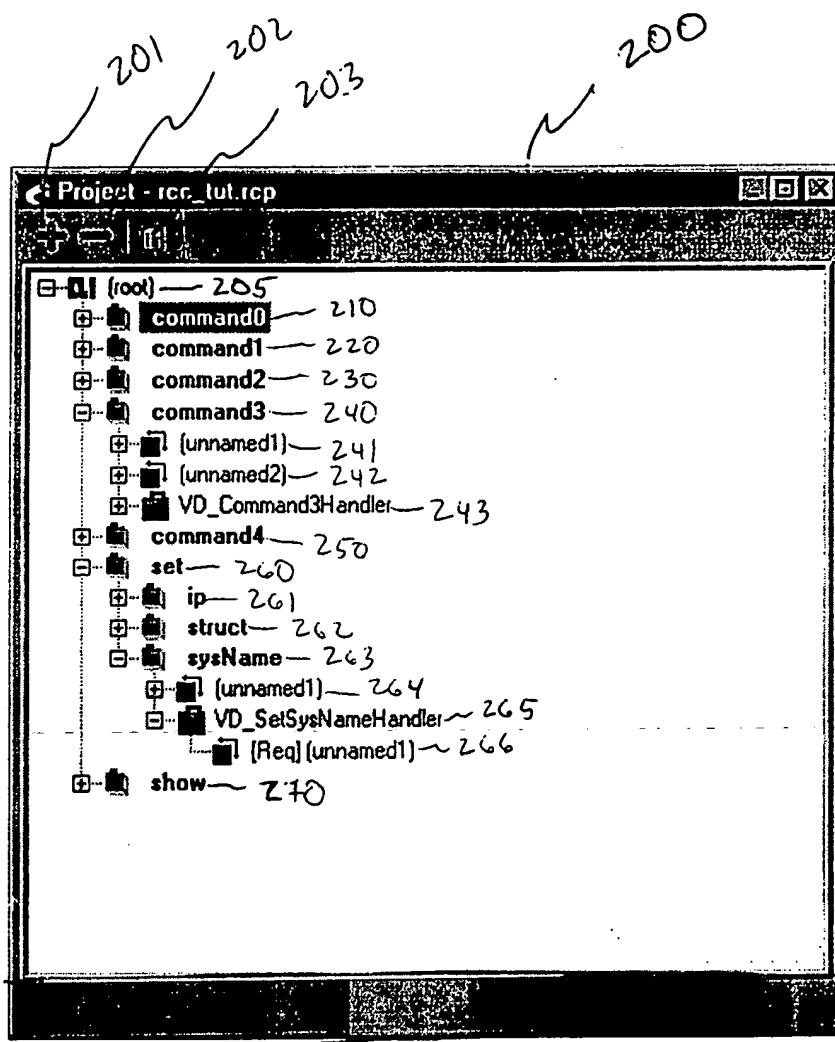


Fig. 4

FIG. 5 is a block diagram of a system architecture 280. The system architecture 280 includes a root node 205, which is connected to five command nodes: command0 210, command1 220, command2 230, command3 240, and command4 250. Command3 240 is further connected to an ip node 261 and a struct node 262. The struct node 262 is connected to a show node 270. The show node 270 is connected to a sysname node 263, which is connected to a Handler Requires node 264, which is connected to a Parm: unnamed1 node 265, which is connected to a SetSysName node 266, which is connected to a Handler Requires node 267, which is connected to a Parm: unnamed1 node 268.

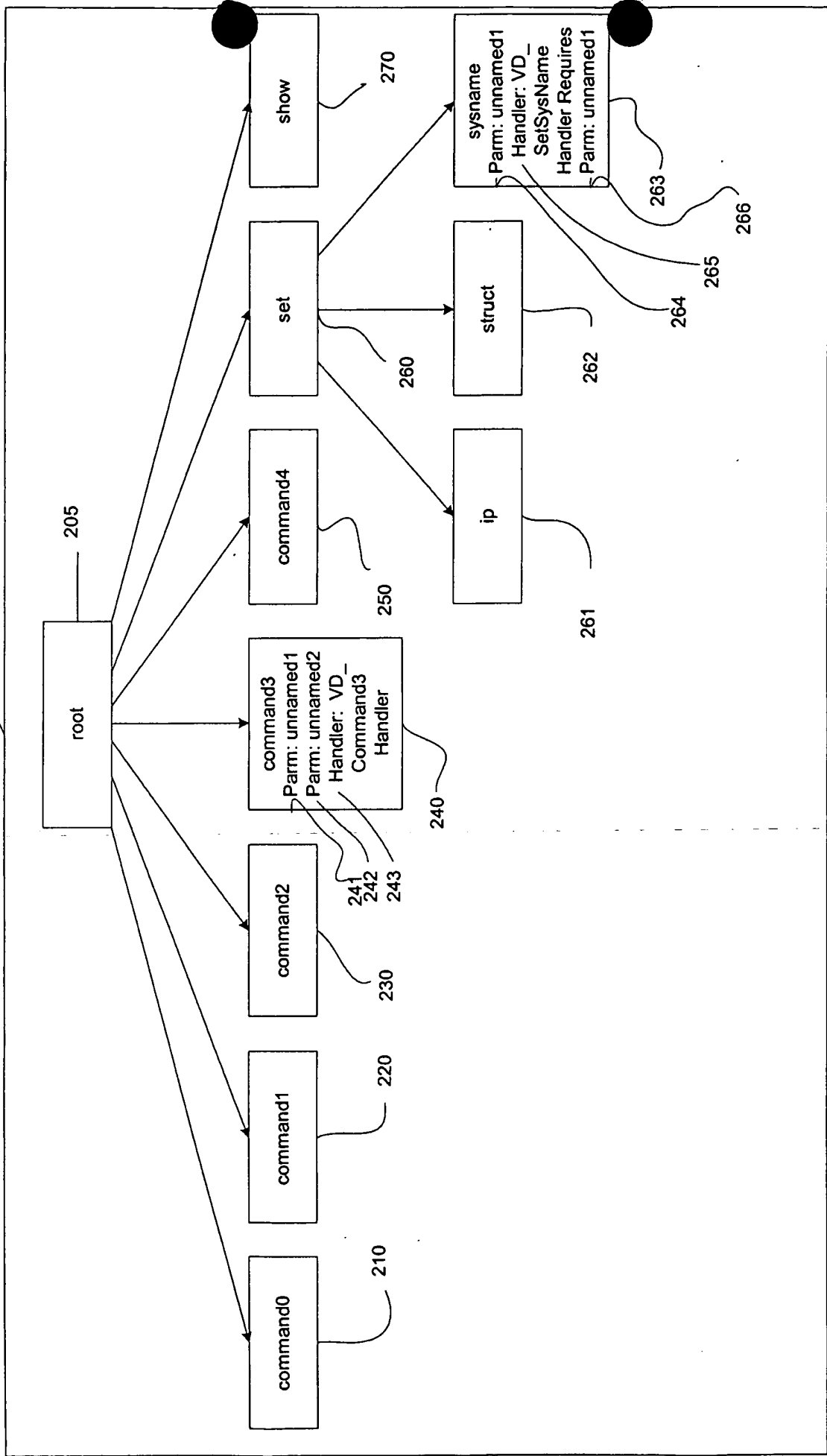


Fig. 5

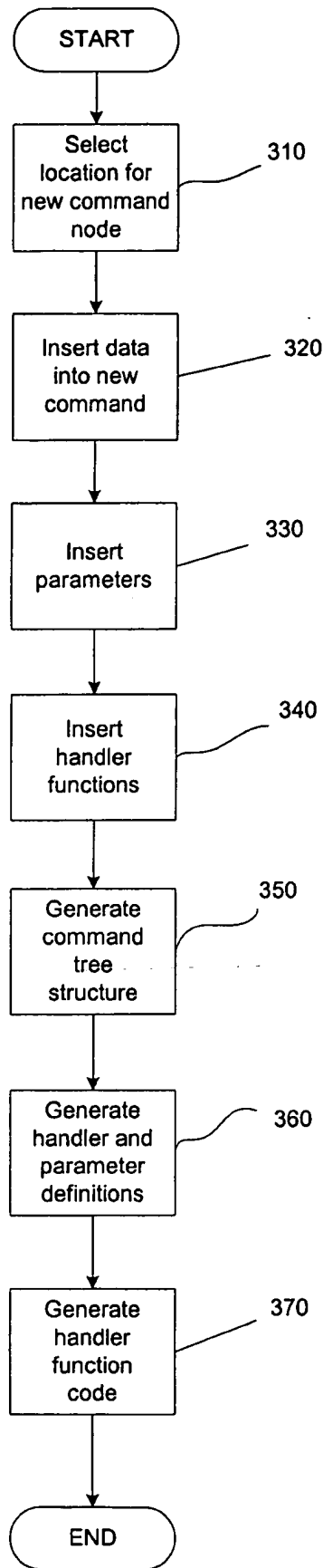


Fig. 6

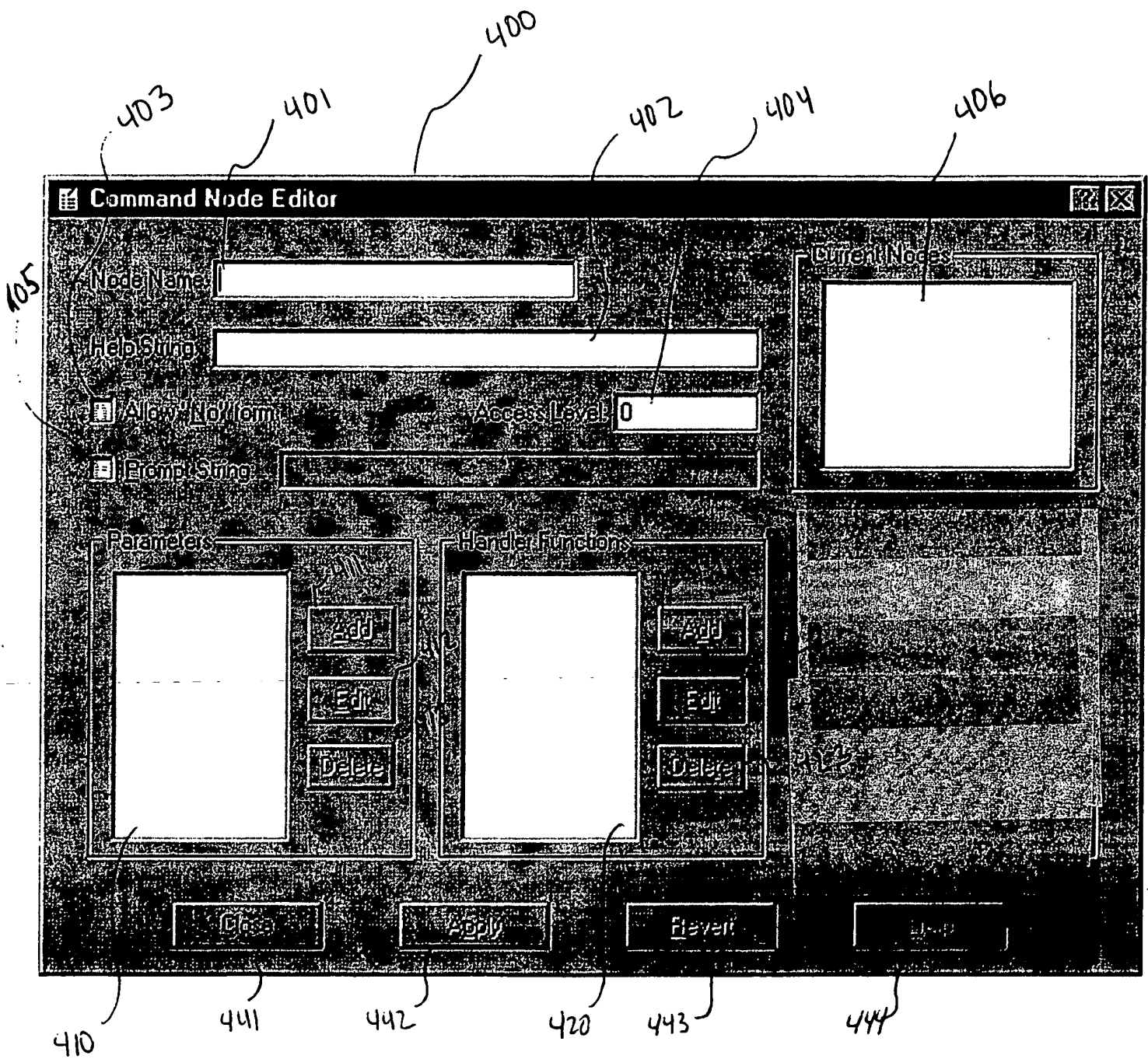


Fig. 7

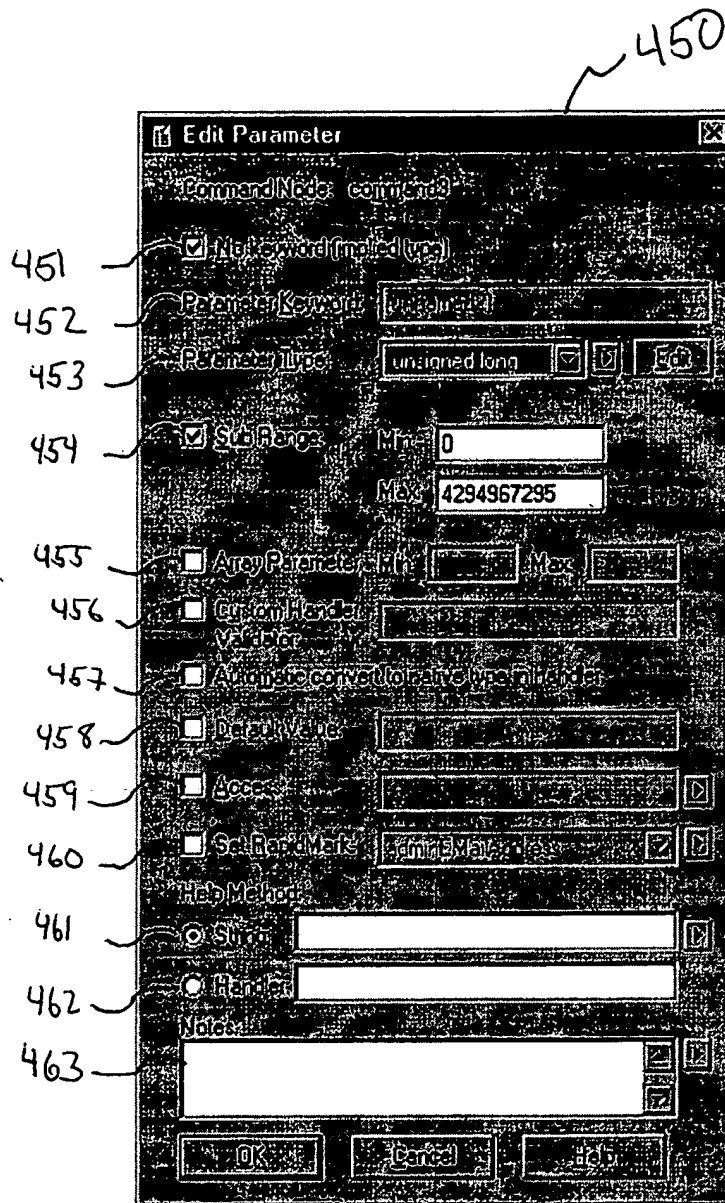


Fig. 8

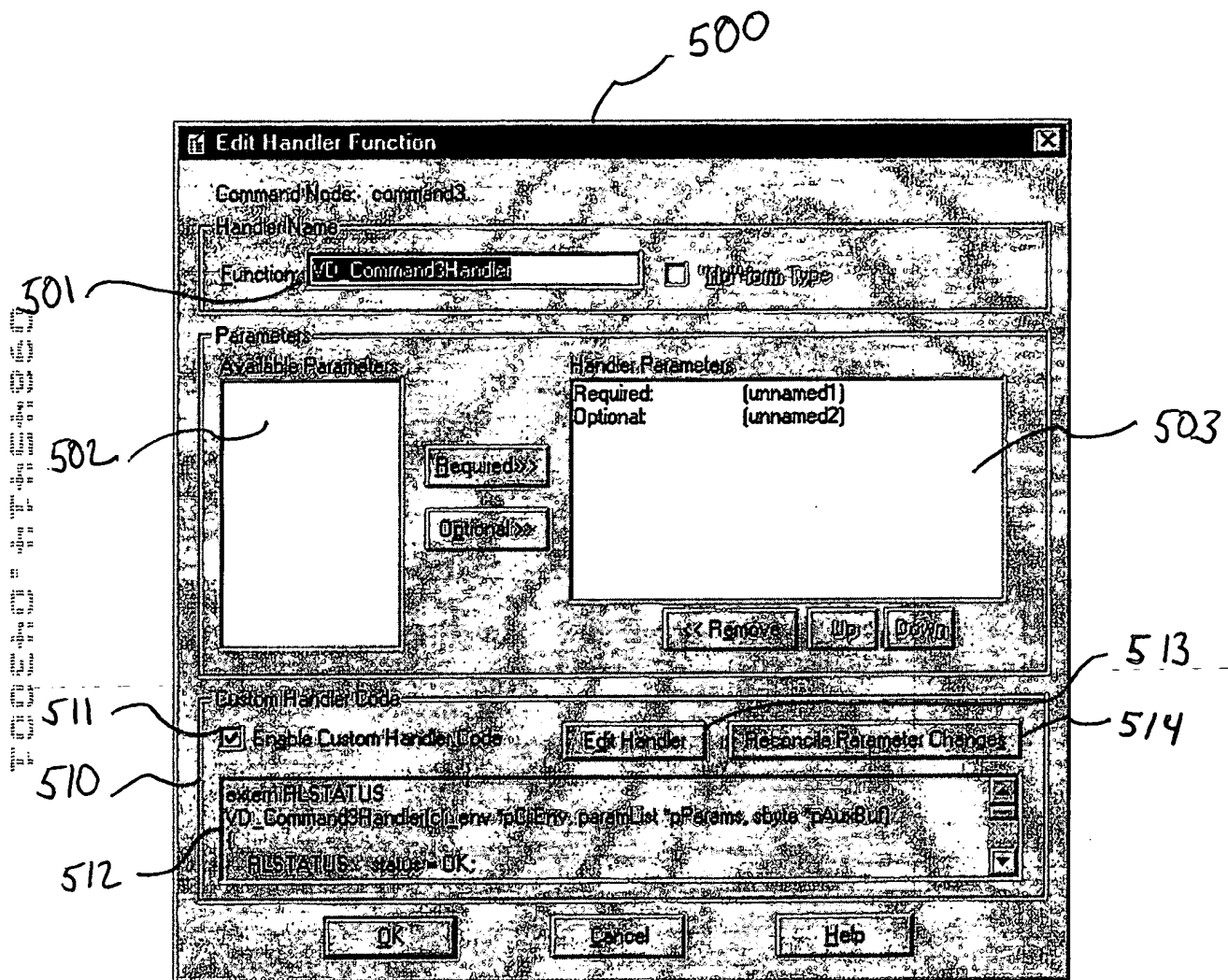


Fig. 9

550

555

560

570

580

590

572

571

583

581

582

584

584

591

592

Quick Command Create

Parent Node: command1

C=command1,HS='help for command1',RM=ipAddress,RM=name

Quick Command Assistant

Insert Command Node

☐ Name (C=)
☐ Help Handler (HH=)
☐ Help Mark (RM=)
☐ Global Node (GL=)
☐ No Generation (NG=)

☐ No Form (NO=)
☐ Access (AC=)
☐ Prompt String (PS=)
☐ Help String (HS=)

Insert Parameter

☐ Name (P=)
☐ Access (AC=)
☐ Validate String (VS=)
☐ Help Mark (RM=)

☐ UnNamed (P=NULL)
☐ Help String (HS=)
☐ Help Handler (HH=)
☐ Note (NT=)

☐ Type (TP=)
☐ Valid Handler (VH=)
☐ Convert Name (C=)
☐ Opt. Param (OP=)

Insert Handler

☐ Name (H=)
☐ No Form (NO=)

☐ Help Param (HP=)

Fig. 10

```

Edit Handler Custom Code
extern RLSTATUS
VD_Command3Handler(cli_env *pCliEnv, paramlist *pParams, sbyte *pAuxBuf)
{
    RLSTATUS    status = OK;
    sbyte       *pVal1;
    paramDescr  *pParamDescr1;
    sbyte       *pVal2 = NULL;
    paramDescr  *pParamDescr2;

    /* get required parameter */
    status = RCC_DB_RetrieveParam(pParams, NULL, nCommand3_NULL1, &pParamDescr1);
    if (OK != status)
    {
        return(status);
    } else pVal1 = (sbyte*)(pParamDescr1->pValue);

    RCC_EXT_WriteStr(pCliEnv, hCommand3MsgRequiredValue);
    RCC_EXT_WriteStrLine(pCliEnv, pVal1);

    /* get optional parameter */
    if (OK == RCC_DB_RetrieveParam(pParams, NULL, nCommand3_NULL2, &pParamDescr2))
    {
        pVal2 = (sbyte*)(pParamDescr2->pValue);
        RCC_EXT_WriteStr(pCliEnv, hCommand3MsgOptionalValue);
        RCC_EXT_WriteStrLine(pCliEnv, pVal2);
    }
}

```

600

Fig. 11

Fig. 12

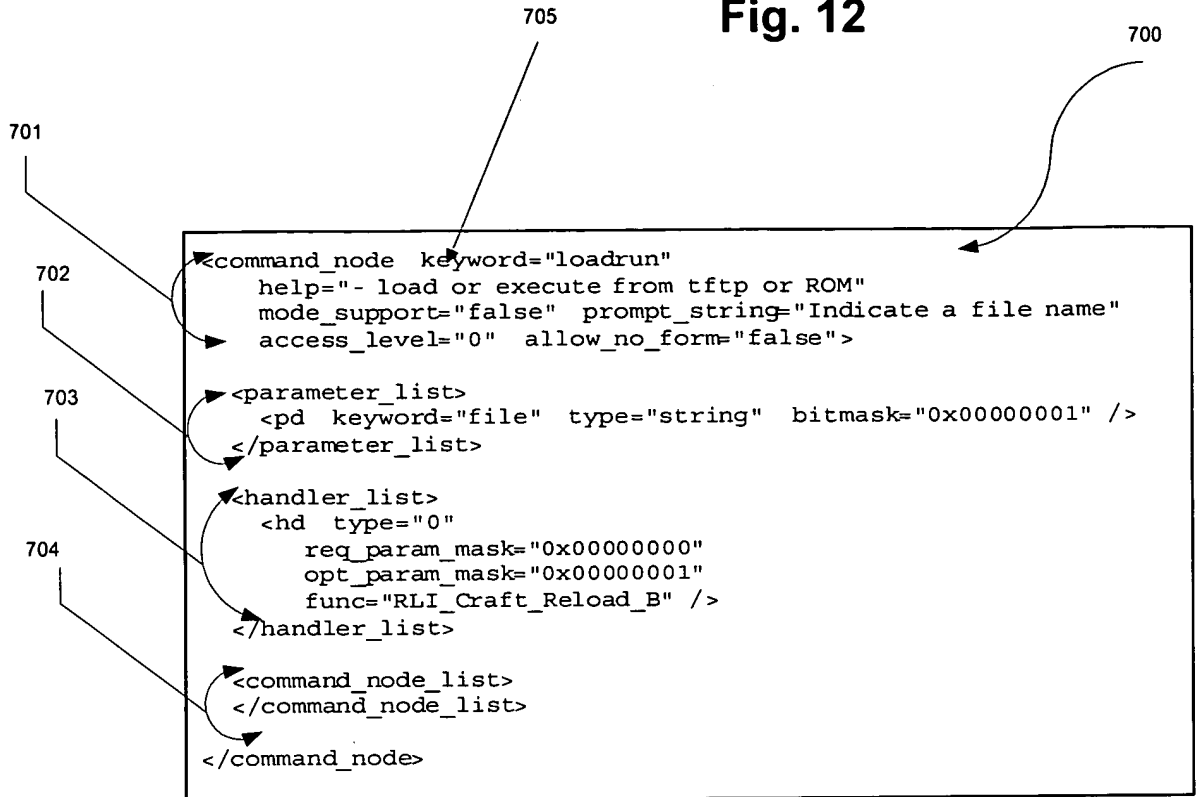


Fig. 13

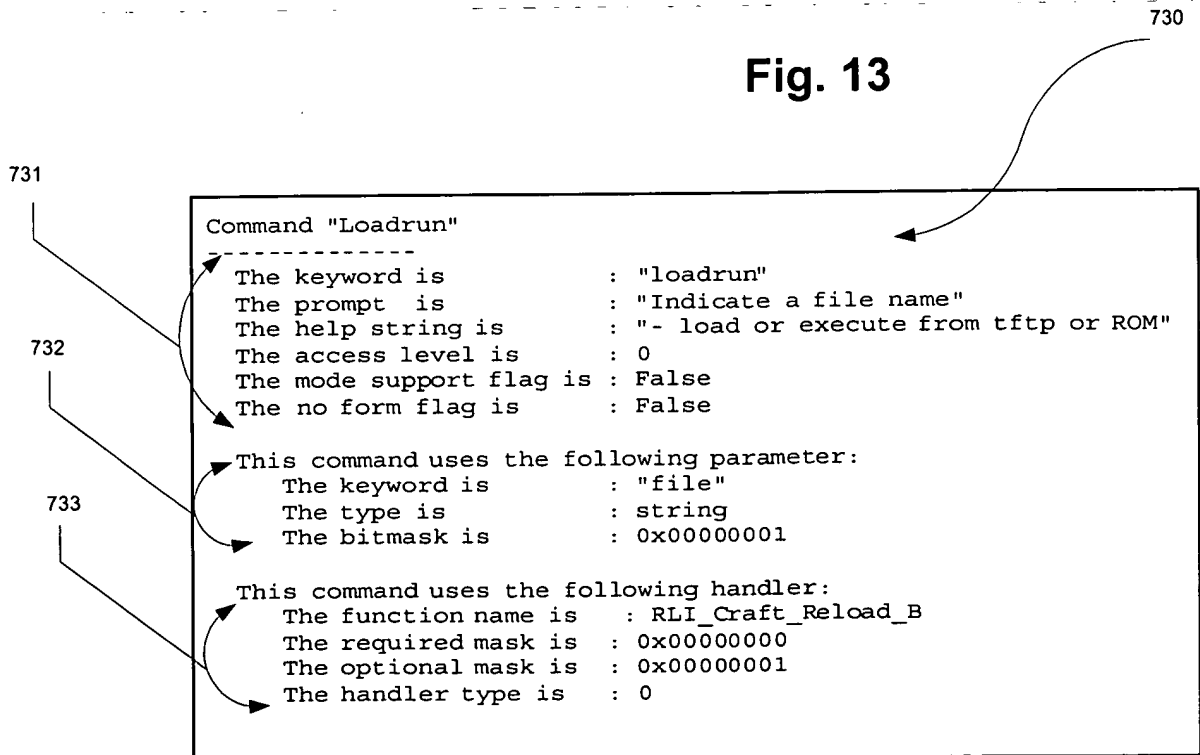


Fig. 14

720

Quick Command Export [X]

Export Selection

721

Insert Command Node

- ☒ Command Name (C=)
- ☒ No Form (NO)
- ☒ Access (AC=)
- ☒ Prompt String (PS=)
- ☒ Help String (HS=)
- ☒ Help Handler (HH=)
- ☒ Rapid Mark (RM=)

Set All
Clear All

722

Insert Parameter

- ☒ Parameter Name (P=)
- ☒ UnNamed (P=NULL)
- ☒ Type (TP=)
- ☒ Valid Handler (VH=)
- ☒ Convert Native (CV)
- ☒ Default Value (DV=)
- ☒ Access (AC=)
- ☒ Validate String (VS=)
- ☒ Rapid Mark (RM=)
- ☒ Help String (HS=)
- ☒ Help Handler (HH=)
- ☒ Notes (NT=)

Set All
Clear All

723

Insert Handler

- ☒ Handler Name (H=)
- ☒ Req. Param (RP=)
- ☒ Opt. Param (OP=)
- ☒ No Form (NO)

Set All
Clear All

Export File:

c:\demo\export.cli

Select File

Export

Cancel

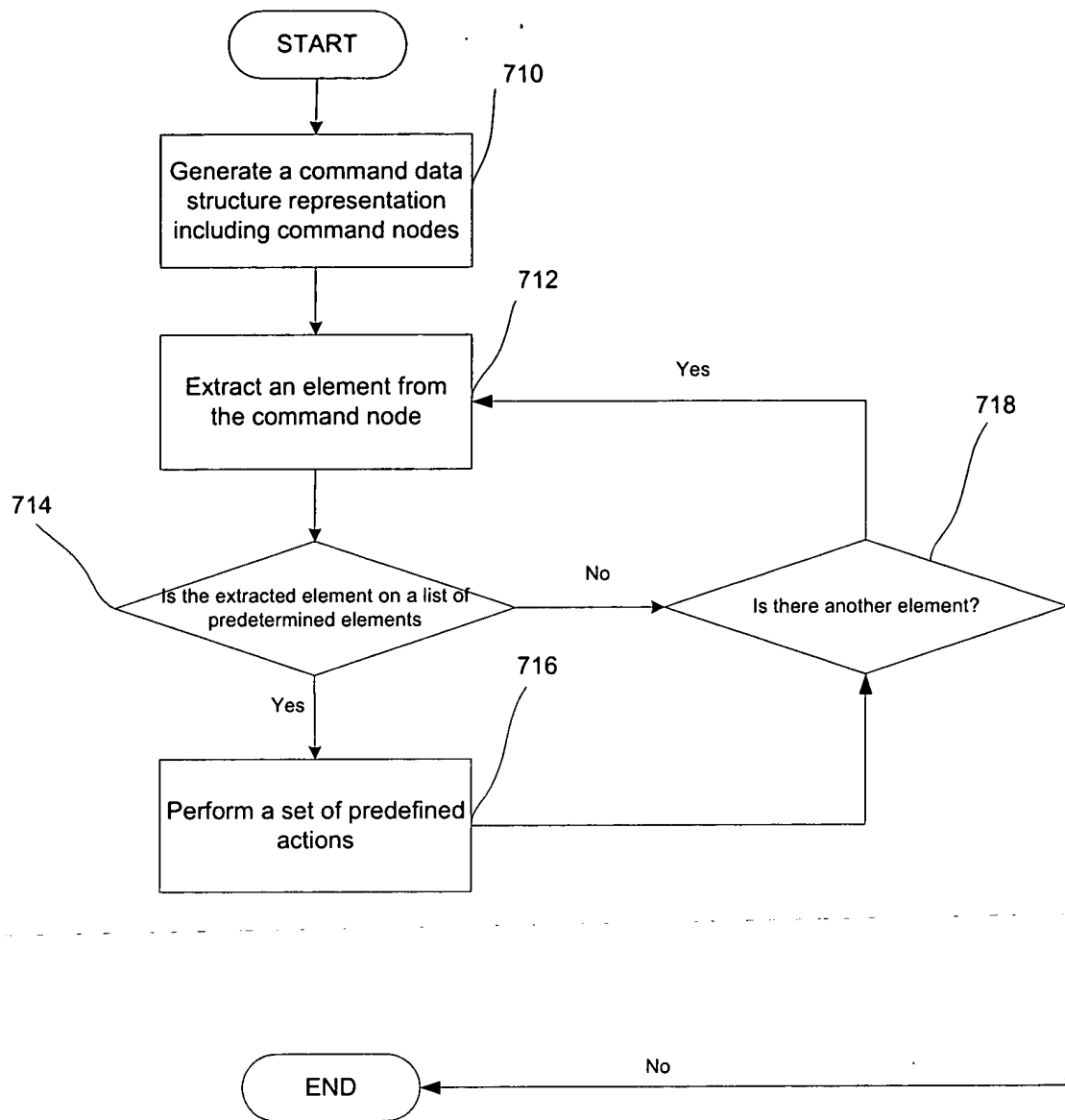


Fig. 15